

# Evolution of D-Studio Middleware technology for BIM integration

Stefan Boeykens, Koenraad Nys  
D-Studio bvba

Abstract: In the evolution towards BIM Level 3 and beyond, D-Studio is developing middleware technology for integration and automation. While initial projects mostly focused on client-specific solutions, the middleware technology is currently evolving into a more flexible and generic technology. The back-end consists of small communication add-ins, which turn desktop BIM applications in live servers, using web technologies to make them accessible for client and other applications. The article discusses current and new solutions, leveraging this approach to produce a more robust and open access to model data, while leveraging native platforms as much as possible.



# Evolution of D-Studio Middleware technology for BIM integration

Stefan Boeykens, Koenraad Nys  
D-Studio bvba

## 1 Intro

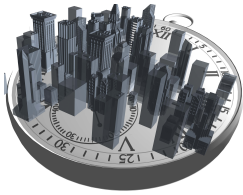
D-Studio is a Belgian consultancy company, focusing on BIM Management, Integration and Innovation. With a small team of architects, engineer and developers, they are developing custom integration tools for clients, but also guiding projects with the BIM Process and at the same time being involved in a wide variety of national and international activities on BIM standardisation and education.

## 2 Context & Problem Statement

The last few years have shown an increased interest in BIM within the construction industry. While the focus, initially, was mostly on the application of BIM in the context of 3D modelling and visualisation, it has shifted significantly towards information management.

Building owners and managers want to have exploitable datasets of the facilities they are maintaining; project developers want quick answers on cost and planning; architects are increasingly being asked to deliver models to the owners and contractors; contractors develop 3D models for cost estimation based on model quantities and construction preparation etc... While in many cases this work relies on the application of popular BIM authoring software, such as Autodesk Revit, Graphisoft ARCHICAD or Trimble Tekla, there are also hybrid usages, with combinations of BIM tools, 3D modelling and CAD (e.g. SketchUp, AutoCAD) and at the same time management and planning environments.

However, many of these systems are hard to properly integrate. While the export and import of IFC models between such systems has significantly improved over the last 5 years and Open BIM support is on the rise, sometimes more direct and automated workflows are required. This article discusses some of the D-Studio technologies and how they solve a wide variety of client demands. At the same time, we reflect in rethinking how many of these individual solutions can evolve into a more uniform and widely applicable technology stack. The focus lies on the technological approach for a middleware solutions that is both flexible, generic and easy to integrate in a variety of BIM systems and platforms.



### 3 Methods & Existing solutions

D-Studio has developed a series of solutions and technologies, which focus specifically on the integration between different tools and environments. They are (or have been) largely developed on customer-demand and often solve a very particular problem in a specific context. Strategically, D-Studio never limited itself to a single BIM system or platform. Different clients, for different reasons, use different systems. For some of them, that may be Revit, but others apply SketchUp or ARCHICAD. Some of them have a custom Access or MS SQL database, whereas others may need an integration into a main ERP system. Sometimes this implies developing add-ins for desktop BIM systems (plug-ins in .NET or C++), whereas other times this involves web and network development or even hybrid solutions, combining desktop, network, web and even graph databases.

Here are a few examples of integrations which have been developed and applied in various projects

#### 3.1 Linking Model with Systems Engineering

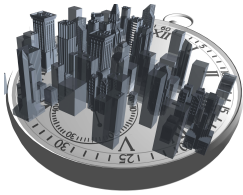
RevRel is a connection between Revit and the Relatics Systems Engineering database. With this add-in, you can collect information directly from the model and map it into the Relatics object-tree, so reports can be generated based on up-to-date information. Such a setup allows to delegate some of the analysis and reporting work to a more suitable environment, while ensuring the model input can be brought up-to-date easily. On a technical level, the SE database is requesting the model via a small server-addin in the BIM software, running on the network and communicating via webservices.

#### 3.2 Linking Model with Spreadsheet

RevCel is another connection, where the BIM model in Revit can be queried from within Microsoft Excel. This also utilises a Server-addin running inside Revit and a plugin to assist the querying process in Excel. This allows collection of elements and attributes, grouped in various worksheets within the workbook. There is the possibility to highlight in realtime the object in the model based on the selected line and to change (certain) parameters.

#### 3.3 Linking Model with other desktop application

BIMOffice is a management software from Abvent France for Bills-of-quantities and cost estimation. The software now integrates with both Revit and ARCHICAD, through means of a custom server-add-in. The BIM software becomes a live database, which can be directly queried from within the BIMOffice environment. Lists of elements, properties and quantities can be queried. There is also element selection directly in the BIM software, so the user can see which elements relate to particular queried results. The use of an http-protocol allows the model tool and the BIMOffice application to stay fairly independent, possibly running on different platforms and on different machines. Through uniformization of the querying API, there is less dependence on the underlying BIM software structure or even version (\* although plug-ins by design are software, platform and version-specific, the underlying code is to a certain extent identical between e.g. the Mac and Windows version of ARCHICAD or between different versions of the BIM software).



### 3.4 Reflection on current integrations

Currently, these integrations focus on the properties, whereas geometry is usually maintained by regular model import (e.g. via IFC or model exports). But there are already examples of encoding certain graphical information, such as a screenshot of the selected object, in a string and store it as a parameter. This allows recreation of this info back again on the receiving side or even to store them with the object.

However, throughout the years many similar problems and similar solutions have emerged in the project portfolio. This ignited the desire to rethink the different approaches and evolve towards a more homogeneous solution, which will capitalise on the different pieces each solving a different part of the overall problem. Most technical hurdles have already been solved in some form, but the reuse of these solutions thus far has proven difficult.

## 4 Next generation of solutions

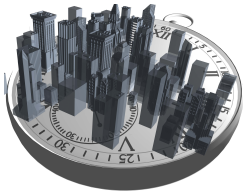
Next we will discuss two cases, where aspects of the overall solution stack are emerging more and more.

### 4.1 Generic integration engine

The back-end of the solution is the generic integration engine. This is a combination of different “layers” of functionality. Most close to the connected BIM system is an add-in, which opens the Desktop application to become more of a Server system. The approach is the launch of a small http server, which provides a uniform access point for the underlying BIM system. It gives access to the model, its elements and their properties. This is mostly used to collect information from the model to a client application, but can be used in the other direction as well. Elements can be selected and/or highlighted in the modelling environment from a request by the client software. Certain changes are also possible, such as names, certain text fields or basic parameters (single-values). More elaborate editing is currently avoided, as this will possible lead to a cascade of changes to not only the selected element, but also all its related elements. For reasons of stability and for clear distinction of permissions, we avoid modifying the underlying element, unless for specific scenarios.

Then there is the communication layer. This can be compared to an HTTP REST access point and closely resembles it, but is currently not strictly written as a fully compliant REST server. Any client sends an http request towards the server running in the BIM system and receives a reply, most commonly in a JSON (Javascript Object Notation) format. This is lightweight when compared to XML and is well supported in a web context.

And then there is a series of “verbs” or request commands, which are a generic way of requesting lists of elements, definitions of property sets and property definitions and also filtering based on element class or category. This series of verbs forms a uniform API. While still tightly related to the underlying BIM system, the verbs are being formulated more generically and the same call can be sent to Revit or ARCHICAD with comparable results (although the underlying returned values still reflect the underlying data model in naming and value types). This includes also the selection of elements and even setting certain parameters, although the comments above still apply.



## 4.2 Building Configurator - a parametric approach for on-demand custom designs

For different clients, there was the request to have a building configurator website, not unlike the way you would configure the options of a car on the reseller catalogue. The underlying model for a project is developed as a parametric assembly of building parts with variations, which are then grouped into a configuration. The different variants are then automatically published as geometric parts, which can be assembled “live” via a web-interface. When a possible buyer for a building, is happy with her configuration for the chosen parcel, she can request an offer, including automatically generated drawings using a prepared template file. The configurator dashboard is presented as an interactive web-application, utilising WebGL to allow realtime 3D in the browser, without the installation of plugins. Performance on the scale of the usually single-family houses is sufficient.

This development should be seen as a client side application of the technology; a combination of using the API for the middleware engine and a front-end to visualise and interact with the information. This also shows how the connection through the API works in two directions: to request information from the dashboard, but also to steer the model generation back to the parametric model to generate a full set of drawings in the native BIM environment.

## 5 Positioning and reflection

### 5.1 What about flux.io?

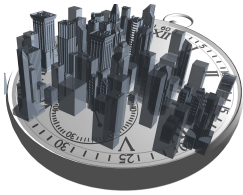
The D-Studio engine approach is obviously not the only way to facilitate automation and interoperability workflows. The flux.io service has gathered quite some attention over the last two years as service which connects different applications in a hybrid desktop-web ecosystem. There is the online flux service which passes parameters between systems, but also through a custom workflow runs in the cloud and is a paid service (with a free tier available for low-volume usage).

And then there are the different application integrations, using plugins for Revit, Excel, SketchUp and Rhino/Grasshopper. The current implementation is directly tied to using the Chrome Browser and only the Windows-platform is supported.

We are currently investigating the main differences in the D-Studio approach and are also looking at possible integration via the Flux and D-Studio API's, which could extend the reach of Flux, but also integrate other systems in a hybrid workflow.

### 5.2 What about IFC?

While this methodology may seem to go against the Open BIM approach, they are more alike than may be assumed at first site. To be able to translate information from one system to another, requires a fairly generic approach to the underlying data models. The same way as the IFC data model describes any construction project, regardless of authoring source or underlying platform or technology, the middleware approach also involves generic translations of projects, objects, property sets and relations. Instead of attempting to enforce everything into a single platform or system, the generic approach often relies on insight of the IFC structure. While currently not implemented, it is also envisaged that the same way a BIM tool can act as “live



data source”, an IFC file may as well be another data source, thus opening the workflows to almost any BIM software available.

One of the reasons that a live interaction may be desired, is the possibility of returning certain results or values into the model and to more easily identify elements, select them, allow the user to return to the native environment to correct issues and update the involved elements. With IFC, the data is a snapshot, but there is no access to the native source model, which would prevent model changes, but even that may be looked at, e.g. by applying “change requests”, for which the use of BCF is currently being investigated by buildingSMART.

## 6 Conclusions and future developments

We have discussed the evolution from a client-driven individual approach towards a more generic and flexible BIM integration approach, solving automation and interoperability challenges our clients are facing today.

At the same time we discussed both technical solutions and inherent limitations with the different pieces of the solution puzzle and how this relates to other solutions or Open BIM.

We see a healthy future where these technologies will enable new workflows, better integration and more independence from the underlying proprietary platforms, while still leveraging the native models as much as possible. This article skipped over more integrated approaches with databases and how they may be embedded and enabled in these workflows.